Boe-Bot Activity

The Boe-Bot robot

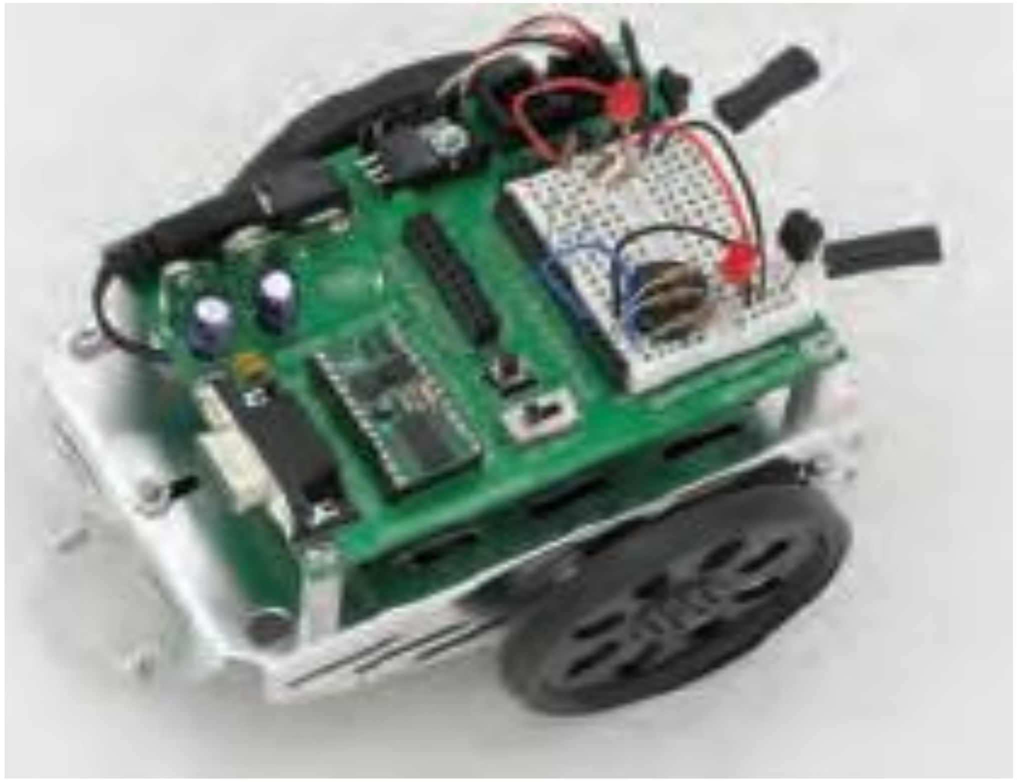
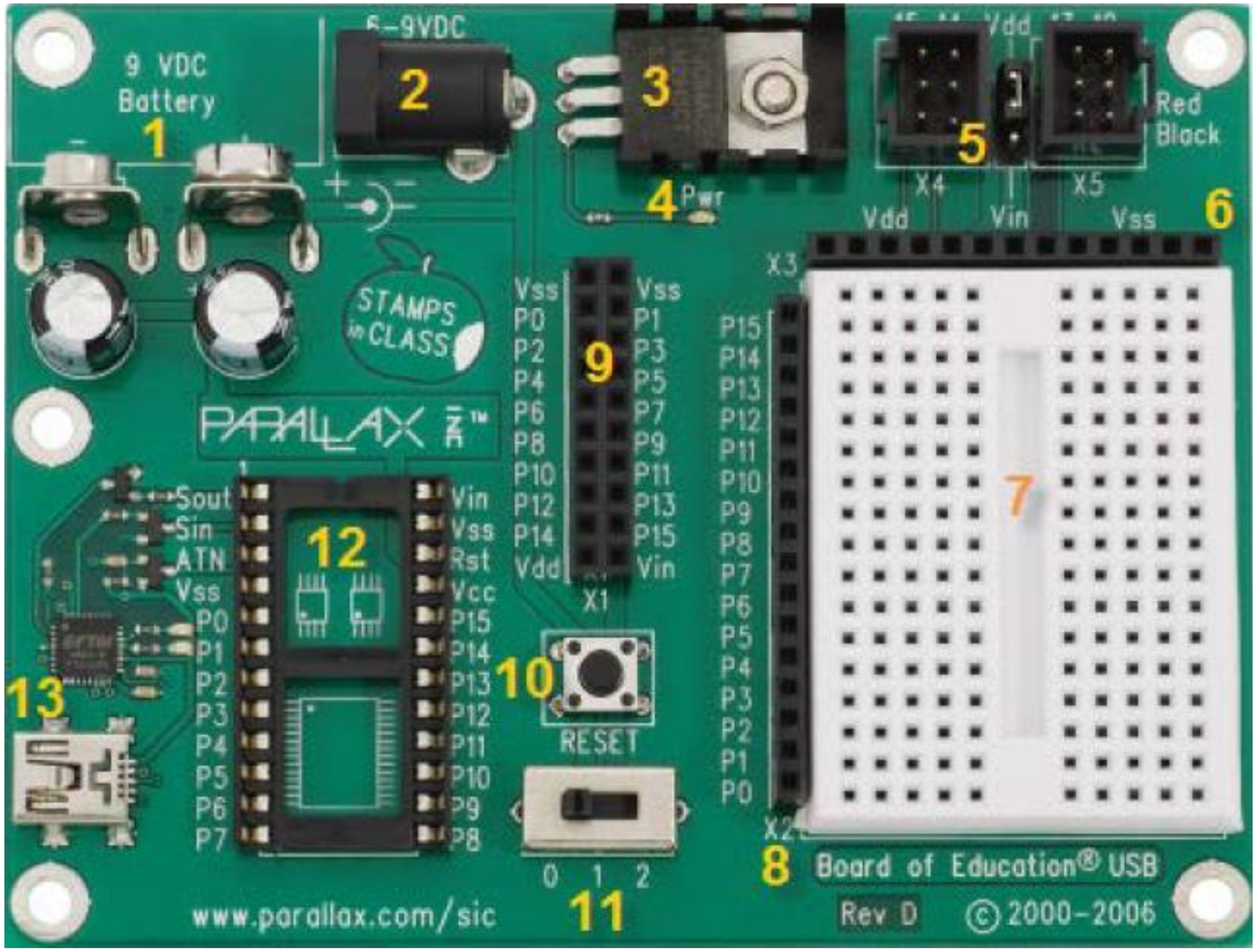


Image source: Robotics with the Boe-Bot —Student Guide

Boe-Bot Circuit Board – Rev D

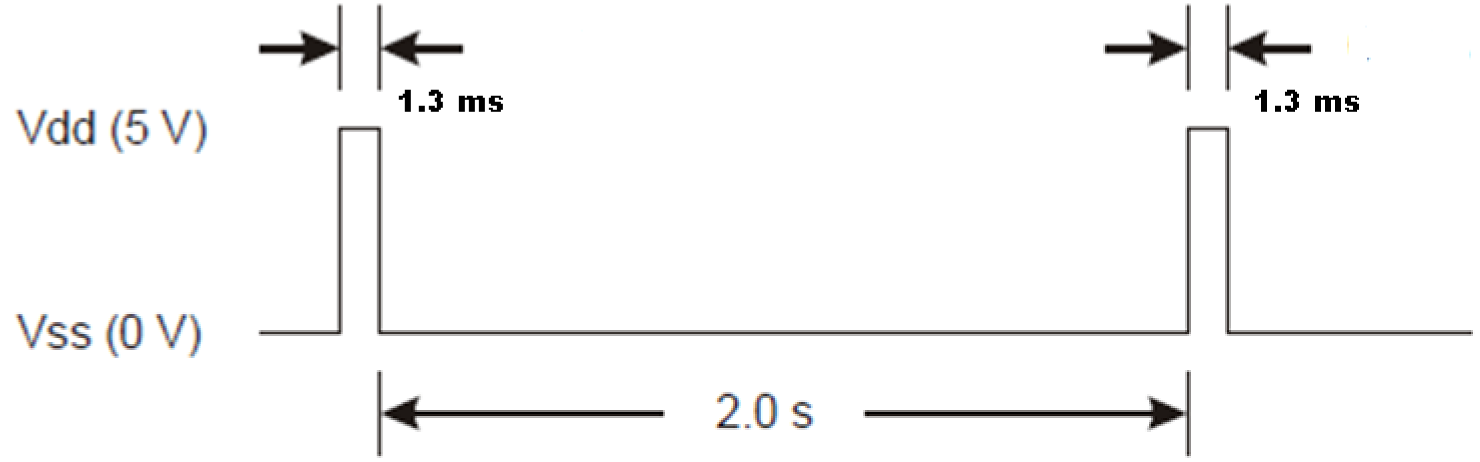


9 VDC Battery
1
6-9VDC
2
3
4 Pwr
5
Red
Black
X4
X5
Vdd
Vin
Vss
6
X3
7
8
Board of Education® USB
Rev D
© 2000-2006
9
X1
Vss P0 P2 P4 P6 P8 P10 P12 P14 Vdd
Vss P1 P3 P5 P7 P9 P11 P13 P15 Vin
P15 P14 P13 P12 P11 P10 P9 P8 P7 P6 P5 P4 P3 P2 P1 P0
X2
10
RESET
11
0 1 2
12
13
Sout Sin ATN Vss P0 P1 P2 P3 P4 P5 P6 P7
Vin Vss Rst Vcc P15 P14 P13 P12 P11 P10 P9 P8
STAMPS in CLASS
PARALLAX
www.parallax.com/sic

Moving the Boe-Bot

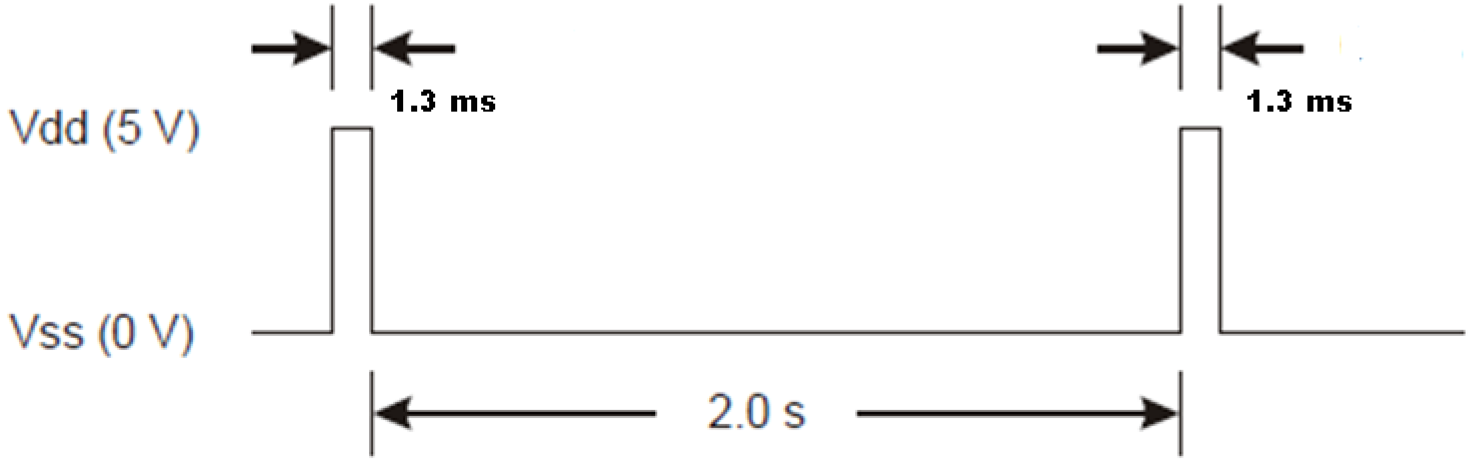
- The Boe-Bot is equipped with servo motors and a microcontroller ("brain"), which can be programmed by the user
- The Boe-Bot's wheel movement is controlled by the rotation of these servo motors
- The servo motors' rotation are, in turn, controlled by electrical ***pulses*** applied to specific circuit **pins** of the Boe-Bot's "brain"
- Each such pin has a number
- Pulses of varying widths can be sent to the pins by the user (you!) via a program →Controlled rotation of servo motors → Controlled movement of the Boe-Bot!

Pulses



- Each HIGH pulse rotates the servo motor
- The direction of rotation is determined by the width (duration) of the HIGH pulse
- Successive HIGH pulses are separated from each other by a PAUSE

Pulses – cont’d



- A HIGH pulse lasting for **1.3 ms** (*milliseconds*) rotates the servo motor **clockwise**

- A HIGH pulse lasting for **1.7 ms** rotates the servo motor **counter-clockwise**

- A HIGH pulse lasting for **1.5 ms** causes the servo motor to **stay still**

How to apply pulses to pins via programming?

- Use the **PULSOUT** command of the PBASIC programming language

- The PULSOUT command's ***argument*** generates a pulse (at pin ***N***) that lasts for ***Argument * 0. 002 milliseconds i.e.***

  Actual duration of pulse = *Argument * 0. 002 milliseconds*

- Rearranging the equation,

  PULSOUT *Argument* = Actual Duration in milliseconds * 500

- Example: The command,

  PULSOUT 12, 650

  generates a HIGH pulse lasting 1.3 ms at pin 12, causing the servo motor connected to pin 12 to rotate full-speed clockwise

Example Code: Moving Boe-Bot for 3 seconds

```
' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Boe-bot is running the program“
counter VAR Byte

FOR counter = 1 TO 122     'Runs the FOR loop 122 times; initially, counter = 1
    PULSOUT 13, 850        'Generates 1.7 ms pulse at pin 13 (pin 13 servo rotates counterclockwise)
    PULSOUT 12, 650        ' Generates 1.3 ms pulse at pin 12 (pin 12 servo rotates clockwise)
    PAUSE 20               'Pause for 20 ms between pulses
NEXT                       'Increment counter by one and repeat loop (repeat till counter = 122)

                             'Time overhead due to the number of instructions in the loop = 1.6 ms

               'TIME TAKEN FOR ONE LOOP  EXECTION = 1.7 + 1.3 + 20 + 1.6 = 24.6 ms

               'TOTAL TIME OF BOE-BOT MOVEMENT= 122 * 24.6 ms =  3 seconds (approx.)
END
```

Exercises

- Read slides 4 through 8

- In the program handouts given, <u>replace the question mark symbols (??) with actual numbers</u> to achieve desired Boe-Bot movement

- Test your answers on Boe-Bot

Exercise 1: Moving Boe-Bot backwards for 5 seconds

```
' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Boe-bot is running the program“
counter VAR Byte

FOR counter = 1 TO ??
    PULSOUT 13, ??
    PULSOUT 12, ??
    PAUSE 20                'Pause for 20 ms between pulses
NEXT                        'Increment counter by one and repeat

                          'Time overhead due to the number of instructions in the loop = 1.6 ms

                  'TIME TAKEN FOR ONE LOOP  EXECTION = 1.7 + 1.3 + 20 + 1.6 = 24.6 ms
END
```

P.S. Nair

Exercise 2: Keeping the Boe-Bot still for 3 seconds

```
' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Boe-bot is running the program“
counter VAR Byte

FOR counter = 1 TO 122
    PULSOUT 13, ??
    PULSOUT 12, ??
    PAUSE 20                     'Pause for 20 ms between pulses
NEXT                            'Increment counter by one and repeat

                            'Time overhead due to the number of instructions in the loop = 1.6 ms

END
```

Exercise 3: Identify the wheel controlled by pin 12

```
' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Boe-bot is running the program“
counter VAR Byte

FOR counter = 1 TO 122
    PULSOUT 13, ??
    PULSOUT 12, ??
    PAUSE 20                    'Pause for 20 ms between pulses
NEXT                          'Increment counter by one and repeat

              'Time overhead due to the number of instructions in the loop = 1.6 ms


END
```

Questions??

Thank you!!